



Tecnológico de Monterrey

Métodos Numéricos

Mtro. Adolfo Centeno T

Differential Equations in Action - Lesson 1

Integrantes:

Maximiliano Macotella Nava, Luz Karen Hernández Hernández, Andres Serrato
Barrera, Julio Rodríguez Salcedo, César David Rosales Álvares, Jim Kevin Holguín
Rodríguez

Lesson 1 - Houston, We Have a Problem

LESSON 1

Lesson 1 - Houston We Have a Problem

[VIEW LESSON](#) →

100% VIEWED

[SHRINK CARD](#)

6. Motion Of The Moon

Lesson 1:
Lesson 1 - Houston We Have a Pro...

SEARCH

RESOURCES

CONCEPTS

- ✓ 1. Welcome
- ✓ 2. Apollo 13
- ✓ 3. What's Important
- ✓ 4. Sine And Friends
- ✓ 5. Trig Practice
- ✓ 6. Motion Of The Moon
- ✓ 7. Derivative Overview

Motion Of The Moon

SEND FEEDBACK

Thanks for completing that!

You got it right!

[CONTINUE](#)

8. Velocity And Acceleration

Lesson 1:
Lesson 1 - Houston We Have a Pro...

SEARCH

RESOURCES

CONCEPTS

- ✓ 8. Velocity And Acceleration
- ★ 9. Equations For Derivatives
- 10. Newton's Second Law
- 11. Supplement: Python Playgro...
- ★ 12. Supplement: Sine and Cosin...
- ★ 13. Surface Gravity
- 14. Forward Euler Method

Velocity And Acceleration

SEND FEEDBACK

Thanks for completing that!

You got it right!

[CONTINUE](#)

[View Intro](#)

[VIEW ANSWER](#)

[SUBMIT ANSWER](#)

9. Equations For Derivatives

The screenshot shows a web interface for a course. On the left is a sidebar with a 'CONCEPTS' list: 9. Equations For Derivatives (checked), 10. Newton's Second Law, 11. Supplement: Python Playground, 12. Supplement: Sine and Cosine with Python (starred), 13. Surface Gravity (starred), 14. Forward Euler Method, and 15. Eulerian Free Fall (starred). The main content area is titled 'Equations For Derivatives' and shows a confirmation modal. The modal has a close button (X), a profile picture of a man, the text 'Thanks for completing that!', a line separator, 'You got it right!', and a 'CONTINUE' button. The background shows handwritten mathematical formulas like $(t+1)^3 - (t+1)^3$ and $(t+1)^2 + \dots$.

10. Newton's Second Law

The screenshot shows a web interface for a course. On the left is a sidebar with a 'CONCEPTS' list: 5. Trig Practice (checked), 6. Motion Of The Moon, 7. Derivative Overview, 8. Velocity And Acceleration, 9. Equations For Derivatives, 10. Newton's Second Law, and 11. Supplement: Python Playground. The main content area is titled 'Trig Practice' and shows a confirmation modal. The modal has a close button (X), a profile picture of a man, the text 'Thanks for completing that!', a line separator, 'You got it right!', and a 'CONTINUE' button. The background shows a right triangle with a 40-degree angle, a hypotenuse of 10, and a list of values: 5.0, 6.4, 6.5, 6.6.

12. Supplement: Sine and Cosine with Python

```
import math
from udacityplots import *

def sin_cos():
    num_points = 50

    x = numpy.zeros(num_points)
    sin_x = numpy.zeros(num_points)
    cos_x = numpy.zeros(num_points)

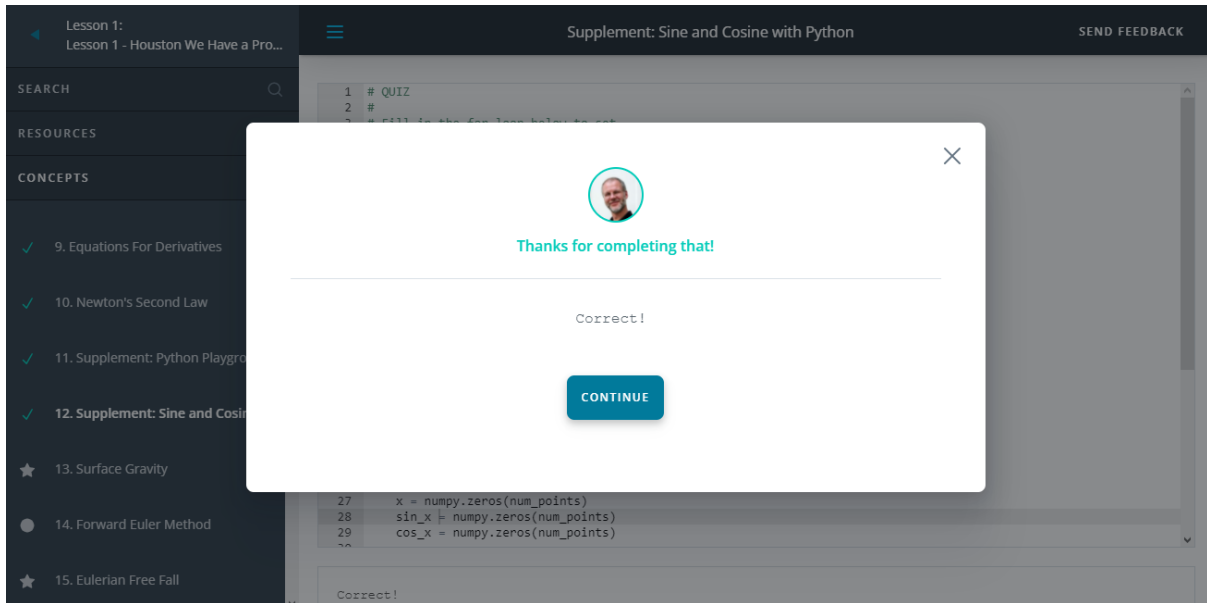
    for i in range(num_points):
```

```

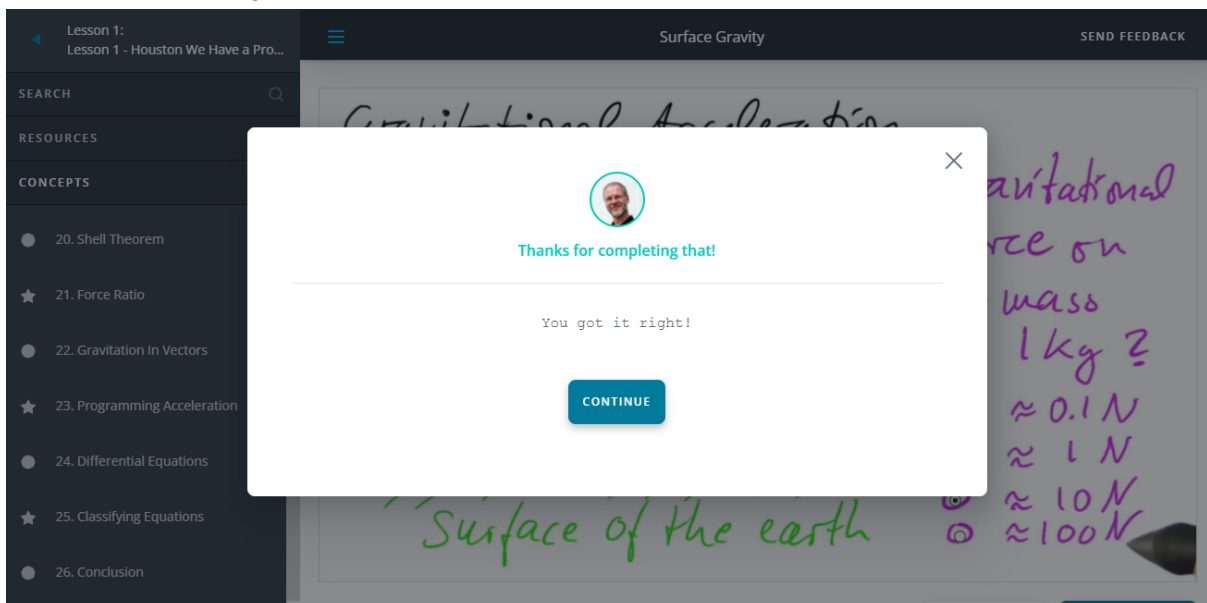
        x[i] = 2. * math.pi * i / (num_points - 1.)
        sin_x[i] = math.sin(x[i])
        cos_x[i] = math.cos(x[i])
    return x, sin_x, cos_x

x, sin_x, cos_x = sin_cos()

@show_plot
def plot_me():
    matplotlib.pyplot.plot(x, sin_x)
    matplotlib.pyplot.plot(x, cos_x)
plot_me()
```



13. Surface Gravity



15. Eulerian Free Fall

```
from udacityplots import *

def forward_euler():
    h = 0.1 # s
    g = 9.81 # m / s^2

    num_steps = 50

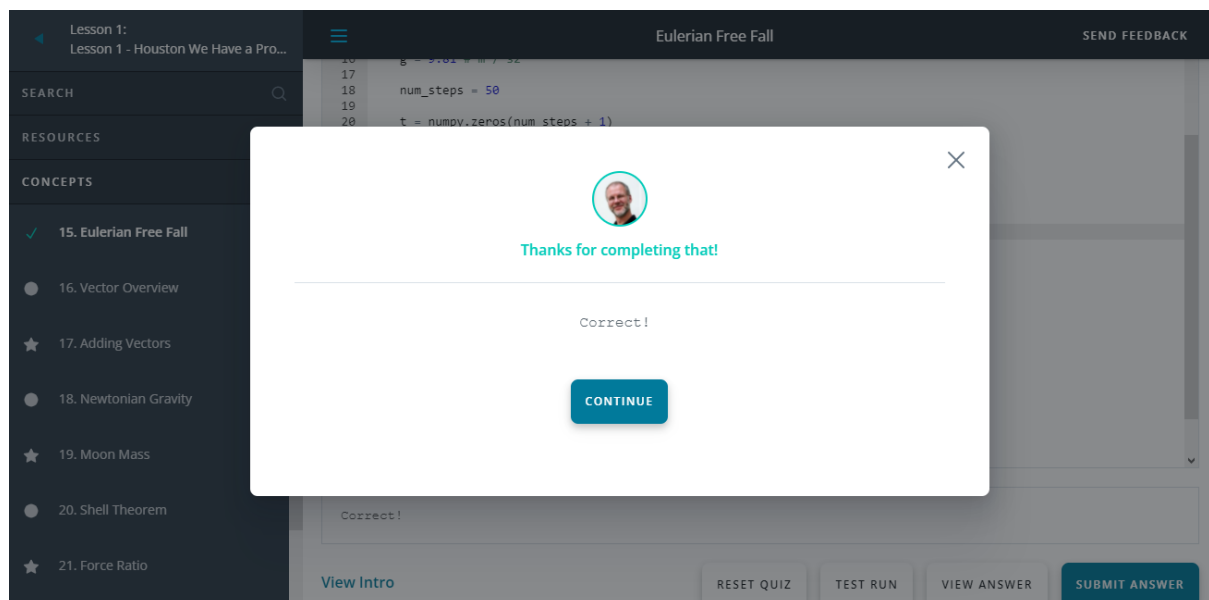
    t = numpy.zeros(num_steps + 1)
    x = numpy.zeros(num_steps + 1)
    v = numpy.zeros(num_steps + 1)

    for step in range(num_steps):
        t[step + 1] = h * (step + 1)
        x[step + 1] = x[step] + h * v[step]
        v[step + 1] = v[step] - h * g
    return t, x, v

t, x, v = forward_euler()

@show_plot # Remove this line when running locally
def plot_me():
    axes_height = matplotlib.pyplot.subplot(211)
    matplotlib.pyplot.plot(t, x)
    axes_velocity = matplotlib.pyplot.subplot(212)
    matplotlib.pyplot.plot(t, v)
    axes_height.set_ylabel('Height in m')
    axes_velocity.set_ylabel('Velocity in m/s')
    axes_velocity.set_xlabel('Time in s')

plot_me()
```



17. Adding Vectors

Lesson 1:
Lesson 1 - Houston We Have a Pro...

Adding Vectors

SEND FEEDBACK

SEARCH

RESOURCES

CONCEPTS

✓ 17. Adding Vectors

● 18. Newtonian Gravity


★ 19. Moon Mass

● 20. Shell Theorem

★ 21. Force Ratio

● 22. Gravitation In Vectors

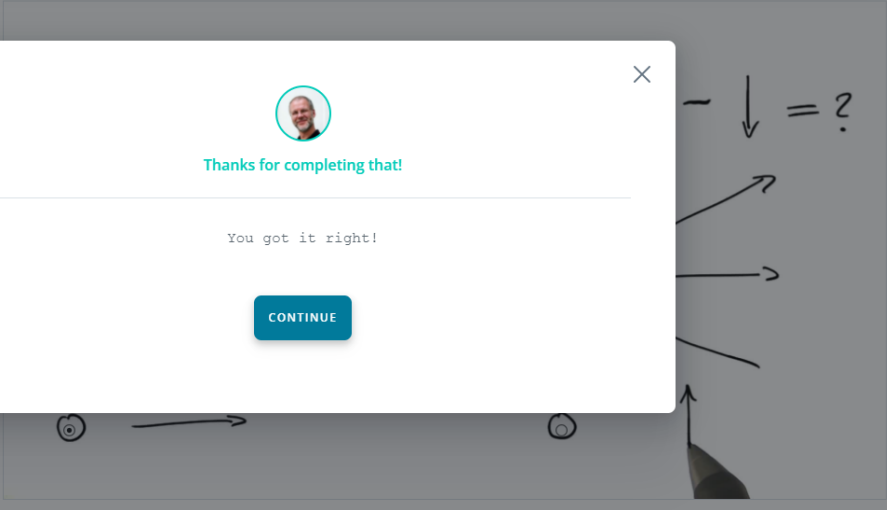
★ 23. Programming Acceleration



Thanks for completing that!

You got it right!

CONTINUE

$$- \downarrow = ?$$


19. Moon Mass

Lesson 1:
Lesson 1 - Houston We Have a Pro...

Moon Mass

SEND FEEDBACK

SEARCH

RESOURCES

CONCEPTS

✓ 19. Moon Mass

● 20. Shell Theorem


★ 21. Force Ratio

● 22. Gravitation In Vectors

★ 23. Programming Acceleration

● 24. Differential Equations

★ 25. Classifying Equations

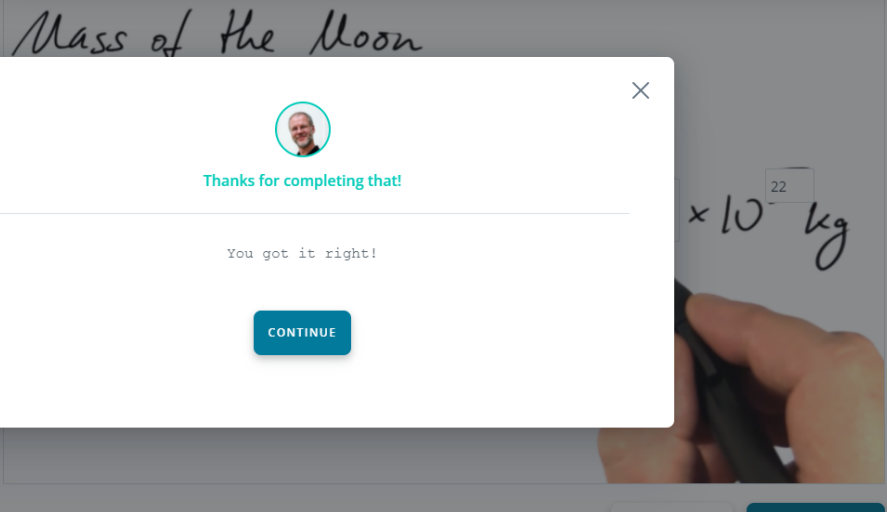


Thanks for completing that!

You got it right!

CONTINUE

Mass of the Moon

$$\times 10^{22} \text{ kg}$$


View Intro

VIEW ANSWER

SUBMIT ANSWER

21. Force Ratio

The screenshot shows a web application interface. On the left is a sidebar with a 'CONCEPTS' list: 20. Shell Theorem (checked), 21. Force Ratio (checked), 22. Gravitation In Vectors, 23. Programming Acceleration (starred), 24. Differential Equations, 25. Classifying Equations (starred), and 26. Conclusion. The main header area displays 'Lesson 1: Lesson 1 - Houston We Have a Pro...' and 'Force Ratio' with a 'SEND FEEDBACK' link. A white modal dialog is centered, featuring a profile picture of a man, the text 'Thanks for completing that!', 'You got it right!', and a 'CONTINUE' button. The background is a blurred image of a chalkboard with mathematical equations.

23. Programming Acceleration

```
import numpy

earth_mass = 5.97e24 # kg
moon_mass = 7.35e22 # kg
gravitational_constant = 6.67e-11 # N m2 / kg2

def acceleration(moon_position, spaceship_position):
    vector_to_moon = moon_position - spaceship_position
    vector_to_earth = - spaceship_position
    return gravitational_constant * (earth_mass /
numpy.linalg.norm(vector_to_earth)**3 * vector_to_earth + moon_mass /
numpy.linalg.norm(vector_to_moon)**3 * vector_to_moon)
```

The screenshot shows the same web application interface as before, but with '23. Programming Acceleration' selected in the 'CONCEPTS' list. The modal dialog now displays 'Correct!' instead of 'Thanks for completing that!'. The background is a blurred image of a document with text and a URL: 'http://forums.udacity.com/cs222/questions/1344/mee-e-book-python-scripting-for-computational-science'. Other visible elements include 'ANSWER' and 'SUBMIT ANSWER' buttons, and a 'NEXT' button at the bottom right.

25. Classifying Equations

Lesson 1:
Lesson 1 - Houston We Have a Pro...

SEARCH

RESOURCES

CONCEPTS

✓ 20. Shell Theorem

✓ 21. Force Ratio

✓ 22. Gravitation In Vectors

✓ 23. Programming Acceleration

✓ 24. Differential Equations

✓ 25. Classifying Equations

● 26. Conclusion

Classifying Equations

SEND FEEDBACK

$\dot{x}(t) = \sin(t)$ $x(t) = \sin(\dot{x}(t))$ $\int \dot{x}(t) = \cos(y(t))$
 $t = (\dot{y}(t))^2$

1

☒


☐

☐

View Intro

VIEW ANSWER

SUBMIT ANSWER



Thanks for completing that!

You got it right!

CONTINUE